

Zastosowanie sieci neuronowych do analizy opinii konsumenckich

Roman Mysan*, Ivan Loichuk, Małgorzata Plechawska-Wójcik

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Niniejszy artykuł przedstawia analizę możliwości zastosowania sieci neuronowych do klasyfikacji danych tekstowych w postaci komentarzy. Ponadto przedstawiono wyniki badania dwóch metod optymalizacji sieci neuronowej: Adam i Gradientu. Celem pracy jest przeprowadzenie badań zachowania się sieci neuronowej w zależności od zmiany parametrów oraz ilości danych użytych do nauczania sieci neuronowej. Na potrzeby realizacji tego celu utworzona została aplikacja testowa korzystająca z sieci neuronowej w celu wyświetlenia ogólnej oceny obiektu noclegowego na podstawie dodanych opinii użytkowników.

Słowa kluczowe: sieć neuronowa; TensorFlow; sztuczna inteligencja

*Autor do korespondencji.

Adresy e-mail: roman.mysan@pollub.edu.pl, ivan.loichuk@pollub.edu.pl

Application of neural networks to the analysis of consumer opinions

Roman Mysan*, Ivan Loichuk, Małgorzata Plechawska-Wójcik

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. This paper presents an analysis of the possibilities of using neural networks to classify text data in the form of comments. Moreover, results of research of two neural network optimization methods: Adam and Gradient are presented. The aim of the work is to conduct research on the behavior of the neural network depending on the change of parameters and the amount of data used to teach the neural network. To achieve the goal, a test application was created. It uses a neural network to display the overall assessment of the accommodation facility based on the added user feedback.

Keywords: neural network; TensorFlow; artificial intelligence

*Corresponding author.

E-mail addresses: roman.mysan@pollub.edu.pl, ivan.loichuk@pollub.edu.pl

1. Wprowadzenie

Sztuczna inteligencja oraz uczenie maszynowe nie są nowymi technologiami. Te technologie od dawna się rozwijają. Niestety dotychczas nie było możliwości tak szerokiego stosowania mechanizmów uczenia maszynowego, ponieważ komputery nie miały odpowiedniej mocy obliczeniowej. Wraz ze wzrostem mocy obliczeniowej komputerów, ten kierunek zdobywa coraz większą popularność. Dzisiaj każdy kto ma komputer i dostęp do Internetu może stworzyć sieć neuronową i testować ją na danych wejściowych. Można również skorzystać z serwisów takich jak na przykład Google lub Amazon, które umożliwiają uruchomienie sieci neuronowych na swoich serwerach. Oprócz mocy obliczeniowej, nie mniej ważnym aspektem jest zestaw danych, na których sieć neuronowa będzie się uczyła. Zestaw danych musi być odpowiednio duży, aby uzyskać wysoką efektywność uczenia.

Organizacje na całym świecie gromadzą duże ilości danych do rozwiązywania różnych problemów biznesowych. Sztuczną inteligencję można zdefiniować jako system, który wykonuje określonego rodzaju zadania przy wykorzystaniu wiedzy, otrzymanej na podstawie interpretowania danych wejściowych. Żeby prawidłowo wykonywać zadania, system uczy się na danych wejściowych, przez interpretowanie danych wejściowych uzyskując wiedzę, którą jest w stanie wykorzystać do wykonania zadań.

Celem badania jest analiza efektywności wykorzystania algorytmów sieci neuronowych w nowoczesnych aplikacjach webowych oraz porównanie dwóch metod optymalizacji sieci neuronowej: Adam i Gradientu. Zadanie zrealizowano w języku programowania Python oraz biblioteki sieci neuronowej TensorFlow.

2. Przegląd literatury

Wraz ze wzrostem dostępności dokumentów elektronicznych i szybkim rozwojem sieci Internet, automatyczna klasyfikacja danych stała się kluczową metodą organizacji danych i odkrywania wiedzy. Właściwa klasyfikacja dokumentów elektronicznych, wiadomości online, blogów i maili wymaga użycia metod wyszukiwania tekstu, uczenia maszynowego oraz technik przetwarzania języka naturalnego w celu uzyskania znaczącej wiedzy. W artykule *A Review of Machine Learning Algorithms for Text-Documents Classification* [4] autorzy poruszają problematykę reprezentacji tekstu oraz omawiają techniki i metody stosowane w klasyfikacji danych tekstowych. Jedną z takich technik jest reprezentacja tekstu (*ang. text representation*), która służy do redukcji złożoności dokumentów i ułatwiania ich automatycznej obsługi. Co do technik nauczania maszynowego to autorzy opisują takie metody nauczania jak: Rocchio's Algorithm, Decision Tree, Decision Rules Classification, Artificial Neural Network,

Genetic Algorithm, Support Vector Machine (SVM), K nearest neighbor (k-NN).

Obecnie istnieje wiele metod optymalizacji sieci neuronowych. Należą do nich: Adagrad, RMSProp, Adam, SGDNesterov, AdaDelta, metoda Gradientu. Użycie metody Adam zostało opisane w artykule *Adam: a method for stochastic optimization* [8]. Autorzy artykułu analizują teoretyczne właściwości konwergencji algorytmu oraz porównują współczynnik zbieżności z najlepszymi znanymi wynikami optymalizacji wypukłej. Wyniki pokazały, że w praktyce Adam działa lepiej i jest korzystniejszy w porównaniu z innymi metodami optymalizacji stochastycznej.

Każda z wymienionych metod jest wspierana przez bibliotekę TensorFlow. Jednym z przykładów zastosowania biblioteki jest użycie sieci neuronowej do przewidywania przepływu ruchu drogowego. Zastosowanie algorytmu zostało opisane przez autorów artykułu *Deep Neural Networks for traffic flow prediction* [5]. Opisana w artykule głęboka sieć neuronowa korzysta z danych o ruchu drogowym w czasie rzeczywistym. Jest to pionierski przykład takich badań. Sugerowany model wykorzystuje rzeczywiste dane ruchu agregowane co pięć minut. Wyniki pokazują, że współczynnik dokładności modelu wynosi około 99%.

Po przeanalizowaniu publikacji naukowych które dotyczą tematu sztucznej inteligencji można stwierdzić, że istnieją publikacje [4, 8] prezentujące wyniki badań wydajności sieci neuronowych przy użyciu różnych metod lub przy zmianie parametrów konfiguracyjnych sieci. W niektórych artykułach naukowych [5] udało się znaleźć przykłady użycia biblioteki TensorFlow w różnych dziedzinach życia.

3. Opis przebiegu badań

Głównym zadaniem aplikacji stworzonej na potrzeby badań jest zbieranie opinii internautów o określonych obiektach noclegowych i generowanie zestawień według określonych kategorii na podstawie analizy przeprowadzanej za pomocą sieci neuronowej. W celu realizacji badań i implementacji sieci neuronowej, zostaną wykorzystane biblioteki: TensorFlow, TensorBoard, NLTK. TensorFlow pozwala wykonywać różnego rodzaju operacje uczenia maszynowego z dużą wydajnością. Wytrenowane modele sieci neuronowej będą zastosowane w aplikacji webowej.

Sieć neuronowa przygotowana na potrzeby badań powinna być tak skonfigurowana i nauczona, aby mogła rozróżnić kategorie (lub kilka kategorii) do której należy komentarz i rodzaj komentarza (pozytywny lub negatywny). Analizując opinie turystów z portalu **booking.com** wybrano cztery kategorie opisu noclegów które najbardziej interesują turystów:

- lokalizacja obiektu noclegowego,
- czystość i komfort,
- personel,
- wyżywienie.

Jednym z najkosztowniejszych, z punktu widzenia nakładu pracy, etapów badania jest zbiór danych do uczenia się.

Podczas przygotowania danych ważna była nie tylko ilość danych, ale i ich jakość. Aby system był gotowy do pracy z dużą ilością danych w postaci opinii użytkowników portalu, informacje były zebrane z największych, dostępnych portali noclegowych. Pozwoliło to na nauczanie sieci na podstawie prawdziwych opinii.

Dane przygotowane na potrzeby uczenia się sieci neuronowej zostały podzielone na 8 kategorii (uwzględniając pozytywne i negatywne opinie dla każdej z czterech kategorii). Pierwszy zestaw danych zawiera około 1700 rekordów, natomiast drugi zestaw zawiera już około 12 tysięcy rekordów.

Aby zbadać wpływ ilości danych na efekt uczenia, sieć neuronowa była trenowana na dwóch zestawach danych. Poza tym, aby sprawdzić jaki wpływ na efekt uczenia ma liczba epok, sieć neuronowa była uczona na 10, 20, 50 oraz 100 epokach.

Zostaną również przeprowadzone badania efektywności metod optymalizacji Adam oraz gradientu w zadaniu klasyfikacji tekstu. Zestaw danych do uczenia będzie obejmował 12 tysięcy komentarzy, zestaw danych testowych będzie obejmował około jednego tysiąca komentarzy.

Wszystkie badania zostaną przeprowadzone z użyciem sieci neuronowej, która posiada jedną warstwę ukrytą z szybkością uczenia 0.001.

Nauczona sieć neuronowa była testowana za pomocą zbioru danych walidacyjnych, niezależnych od zbioru uczącego. Ilość danych testowych jest dosyć duża, aby uzyskać dokładne wyniki testowania.

Podczas analizy efektywności uczenia sieci neuronowych zostaną uwzględnione: wartość odchylenia w procesie uczenia, wartość odchylenia w wyniku końcowym oraz dokładność otrzymaną po przetestowaniu nauczanej sieci neuronowej.

4. Metody optymalizacji gradientu oraz Adam

Do optymalizacji wartości odchylenia w sieciach neuronowych najczęściej wykorzystuje się metodę gradientu, która przy każdej iteracji algorytmu, szuka kierunku, w którym wartość odchylenia maleje. Wadą klasycznej metody gradientu jest to, że podczas procesu uczenia sieci neuronowej (optymalizacji wartości odchylenia) wykorzystuje stałą wartość parametru szybkości uczenia przez cały proces uczenia, co przy małej liczbie epok może spowodować nie satysfakcjonujący wynik.

Aby rozwiązać problem metody gradientu, Diederik P. Kingma oraz Jimmy Lei Ba opracowali metodę optymalizacji o nazwie Adam i opisali ją w swojej pracy: „Adam: a method for stochastic optimization” [2]. Metoda optymalizacji Adam ma wysoką wydajność i jest bardzo efektywna w optymalizacji. Główną zaletą tej metody jest to, że oblicza ona wartość parametru szybkości uczenia do każdej wagi sygnału, który łączy się z neuronem i podczas uczenia sieci Adam może odpowiednio dopasowywać tą wartość do każdego sygnału.

5. Prezentacja rezultatów badań

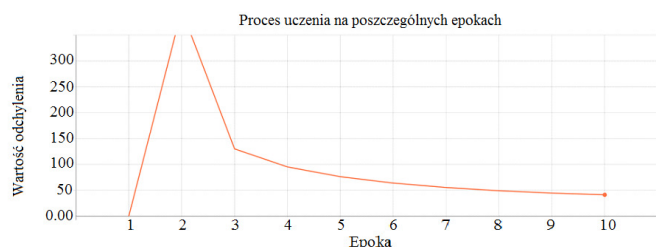
5.1. Badanie 1

Z zestawem danych, który obejmuje 1700 komentarzy, otrzymano dokładność 42%, a wartość odchylenia udało się zoptymalizować do 38. Wyniki uczenia przedstawiono w tabeli 1. Patrząc na otrzymaną dokładność można stwierdzić, że wynik nie jest zadowalający, ale z takim małym zestawem danych trudno otrzymać lepszy. Dane muszą być wysokiej jakości i posiadać dużo cech (ang. *features*), na których sieć będzie się uczyła klasyfikować komentarze. Poza tym, analizując otrzymaną wartość odchylenia (38), można stwierdzić, że 10 epok uczenia było wartością zbyt małą.

Tabela 1. Wyniki procesu uczenia w Badaniu 1

Wartość odchylenia	Dokładność (%)
38,88	42

Analizując proces uczenia, który przedstawiono na rysunku 1, można zauważyć duży skok w wartości odchylenia pomiędzy pierwszą a drugą epoką, ale po 2 epoce następuje duży spadek wartości odchylenia. Podczas całego procesu uczenia sieć rozwijała się, ale po 9 epoce widać, że progres w uczeniu zmniejszył się.



Rys. 1. Proces uczenia na poszczególnych epokach

Na potrzeby analizy wpływu liczby epok na efekt uczenia, liczbę epok było zwiększono kilka razy: do 20 epok, do 50 epok i do 100 epok. Wynik uczenia przedstawiono w tabeli 2.

Tabela 2. Wynik uczenia sieci dla poszczególnych liczby epok

Liczba epok	Wartość odchylenia	Dokładność (%)
10	38,80	42
20	32,35	42
50	31,45	47
100	31,37	47

Analizując wyniki z tabeli 2 otrzymane po zwiększeniu liczby epok, można stwierdzić, że zwiększenie liczby epok miało pozytywny wpływ na efekt uczenia. Porównując wynik dokładności, który otrzymano po 10 epokach uczenia z wynikiem otrzymanym po 100 epokach widać, że dokładność wzrosła o 5%. Wartość odchylenia udało się zoptymalizować z 38 do 31. Poza tym, analizując pozostałe wyniki, można zauważyć, że po uczeniu, które trwało 50 epok otrzymano prawie taki sam wynik, jak po uczeniu przy uczeniu po 100 epokach. W tym przypadku wartość odchylenia zmniejszyła się o bardzo małą wartość, wzrost dokładności jest niewielki.

Podsumowując można stwierdzić, że liczba epok w tym badaniu nie miała dużego wpływu na wynik końcowy, a optymalną liczbą epok do uczenia jest 50. 100 epok jest natomiast zbyt dużą liczbą, bo wynik końcowy prawie się nie różni od wyniku uczenia po 50 epokach.

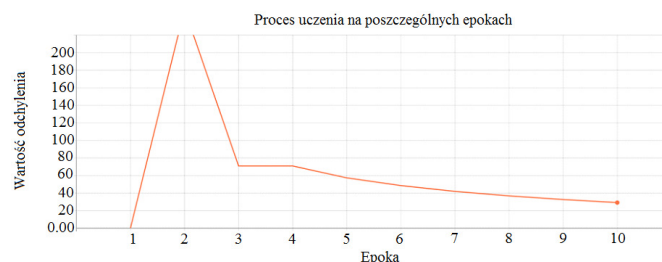
5.2. Badanie 2

Aby zbadać wpływ ilości danych na efekt uczenia, następne badania były przeprowadzone z drugim zestawem danych (który obejmuje 12 tysięcy komentarzy). Wynik uczenia z tym zestawem danych przedstawiono w tabeli 3. Analizując wynik widać, że dokładność po 10 epokach uczenia stanowi 58%, a wartość odchylenia udało się zoptymalizować do 26. Porównując otrzymany wynik z wynikiem badania 1 widać, że wraz z większym zestawem danych otrzymano o wiele lepszy efekt uczenia. Po zwiększeniu ilości danych dokładność wzrosła o 16%, a wartość odchylenia zmniejszyła się o 12.

Tabela 3. Wynik uczenia sieci w badaniu 2

Wartość odchylenia	Dokładność (%)
26,19	58

Analizując proces uczenia, który przedstawiono na rysunku 2, można zauważyć, że wykres jest bardzo podobny do wykresu dla badania 1 (rysunek 1). Jednak w badaniu 1 zauważono, że po 9 epokach progresu w uczeniu prawie nie było, natomiast w tym przypadku dobrze widać, że podczas całego procesu uczenia sieć neuronowa ma progres w optymalizacji wartości odchylenia. Aby uzyskać jeszcze lepszy efekt uczenia, liczba epok musi być zwiększona.



Rys. 2. Proces uczenia na poszczególnych epokach

Poprzednio stwierdzono, że efekt uczenia z większym zestawem danych może być lepszy po zwiększeniu liczby epok. Aby zweryfikować tą potencjalną zależność, liczbę epok zwiększono kilka razy. Wyniki przedstawiono w tabeli 4.

Tabela 4. Wynik uczenia sieci dla poszczególnych liczby epok

Liczba epok	Wartość odchylenia	Dokładność (%)
10	26,19	58
20	9,77	61
50	3,07	64
100	2,29	64

Analizując wyniki można stwierdzić, że w tym przypadku zwiększenie liczby epok również miało pozytywny wpływ na efekt uczenia jak i w badaniu 1. Zwiększenie liczby epok spowodowało wzrost dokładności o 6% i duży spadek wartości odchylenia. Patrząc na wyniki w tabeli 4 można

zauważyć, że maksymalnie wysoką dokładność udało się otrzymać po 100 epokach uczenia, minimalną wartość odchylenia też otrzymano po 100 epokach uczenia. Poza tym, analizując pozostałe wyniki można zauważyć, że otrzymana dokładność przy uczeniu, które trwało 50 epok nie różni od dokładności przy uczeniu, które trwało 100 epok. Sytuacja jest bardzo podobna jak w badaniu 1, gdy po 50 epokach otrzymane wyniki były podobne jak te uzyskane przy 50 epokach.

Z danego badania wynika, że zwiększenie liczby epok miało pozytywny wpływ na efekt uczenia, a optymalną liczbą epok jest 50.

5.3. Badanie 3

Wynik uczenia sieci neuronowej z wykorzystaniem metody gradientu przedstawiono w tabeli 5.

Tabela 5. Wynik uczenia z użyciem metody gradientu dla poszczególnej liczby epok

Liczba epok	Wartość odchylenia	Dokładność(%)
10	209,21	15
20	157,49	27
50	145,12	29
100	112,08	38

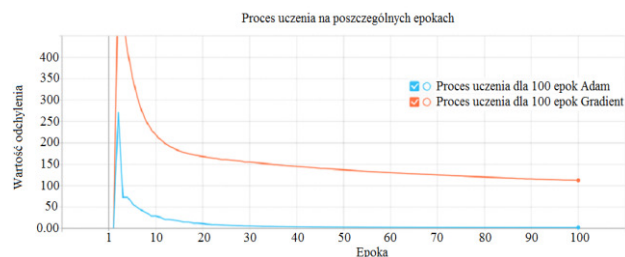
Analizując wynik można stwierdzić, że sieć w procesie uczenia cały czas się doskonaliła, bo wyniki pokazują, że dokładność cały czas rosła, a wartość odchylenia zmniejszała się. W porównaniu z wynikami, które otrzymano przy uczeniu sieci z wykorzystaniem metody Adam, które przedstawiono w tabeli 6, można zauważyć, że podczas uczenia z wykorzystaniem metody Adam uzyskano o wiele lepszy wynik klasyfikacji.

Tabela 6. Wynik uczenia z użyciem metody Adam dla poszczególnej liczby epok

Liczba epok	Wartość odchylenia	Dokładność(%)
10	26,19	58
20	9,77	61
50	3,07	64
100	2,29	64

Dokładność, którą uzyskano przy uczeniu sieci neuronowej za pomocą algorytmu Adam jest o wiele większa w porównaniu z dokładnością uzyskaną podczas uczenia z wykorzystaniem metody gradientu. Dokładność, po 10 epokach z wykorzystaniem metody gradientu stanowi tylko 15%, w porównaniu do dokładności otrzymanej metodą Adam, która stanowi 58%. Duża różnica jest nie tylko w dokładności, ale również w wartości odchylenia, którą otrzymano w wyniku uczenia sieci. Wartość odchylenia po 10 epokach z wykorzystaniem metody gradientu stanowi 209, natomiast metoda Adam zoptymalizowała wartość odchylenia do 26. Analizując kolejne wyniki uczenia, po zwiększeniu liczby epok z 10 do 20, widać, że dokładność przy uczeniu z wykorzystaniem metody Adam wzrosła o 3%, ale patrząc na wynik, który uzyskano przy pomocy metody gradientu, widać duży wzrost dokładności, z 15% do 27%. Dokładność nadal

jest o wiele niższa w porównaniu do otrzymanej przy pomocy metody Adam. Natomiast, analizując wartość odchylenia po zwiększeniu liczby epok, można zauważyć duży spadek wartości odchylenia w obu przypadkach. Warto zauważyć, że po 50 epokach metoda Adam straciła progres w optymalizacji wartości odchylenia, wówczas progresu w dokładności też nie było. Porównując te wyniki do tych uzyskanych z wykorzystaniem metody gradientu, można zauważyć, że metoda gradientu cały czas miała progres.



Rys. 3. Porównanie wartości odchylenia

Analizując wykres na rysunku 3, który ilustruje proces uczenia dla obu metod, widać, że metoda Adam uzyskiwała optymalną wartość odchylenia dużo wcześniej niż metoda gradientu. Po 30 epokach, metoda Adam prawie nie optymalizowała wartość odchylenia. W przypadku metody gradientu, cały czas widać było progres, ale wartość odchylenia była o wiele większa niż dla metody Adam. W wyniku uczenia sieci metoda Adam uzyskiwała optymalną wartość odchylenia, ale dla metody gradientu epok było za mało, aby maksymalnie zoptymalizować wartość odchylenia.

Podsumowując, metoda Adam okazała się o wiele lepsza niż metoda gradientu. Metoda gradientu uzyskała o wiele gorsze wyniki uczenia niż metoda Adam przy takiej samej liczbie epok. Oczywistym jest fakt, że dla metody gradientu epok było za mało. Natomiast, dla metody Adam liczba epok była wystarczająca, aby uzyskać optymalną wartość odchylenia oraz dokładność. Poza tym, metoda Adam pokazała wysoką dokładność oraz niską wartość odchylenia już po 10 epokach uczenia, gdy metoda gradientu dawała o wiele gorsze wyniki.

6. Wnioski

Po przeanalizowaniu wyników badań 1 oraz 2 można stwierdzić, że zestaw danych ma duży wpływ na efekt uczenia. Poza tym, na efekt uczenia ma wpływ nie tylko ilość danych do uczenia, ale i jakość tych danych. Porównując wyniki badania 1 oraz 2 było widać, że wraz ze zwiększeniem danych do uczenia dokładność wzrosła, natomiast 64% dokładności nie jest wynikiem satysfakcjonującym. Takie zachowanie sieci jest spowodowane słabą jakością danych w drugim zestawie danych (który obejmuje 12 tysięcy komentarzy). Poza tym badania pokazały, że liczba epok ma duży wpływ na wyniki uczenia, dlatego, liczbę epok trzeba dopasowywać, aby uzyskać jak najlepsze wyniki uczenia.

Badanie 3 miało na celu porównanie metod optymalizacji Adam oraz gradientu. Z otrzymanych wyników badania 3 widać, że metoda optymalizacji Adam daje wysoką wydajność i jest o wiele lepsza od klasycznej metody gradientu. Przy uczeniu z wykorzystaniem metody

optymalizacji Adam udało się uzyskać optymalną wartość odchylenia o wiele wcześniej w porównaniu do uczenia z wykorzystaniem metody gradientu. Dla metody gradientu epok było za mało, natomiast dla metody Adam epok było wystarczająco (nawet za dużo), aby uzyskać optymalną wartość odchylenia.

Literatura

- [1] M. Fábio, M. F. Alan, Neural network programming with Java, Packt Publishing, 2016
- [2] P. D. Kingma, J. Lei Ba: Adam: a method for stochastic optimization, Published as a conference paper at ICLR 2015
- [3] B. Yoshua, Practical Recommendations for Gradient-Based Training of Deep Architectures, Version 2, Sept. 16th, 2012
- [4] K. Aurangzeb, B. Baharum, L. Lam Hong*, K. Khairullah, A Review of Machine Learning Algorithms for Text-Documents Classification, Journal of advances in information technology, vol. 1, no. 1, february 2010
- [5] Y. Hongsuk, J. HeeJin, B. Sanghoon, Deep Neural Networks for traffic flow prediction, Published in: 2017 IEEE International Conference on Big Data and Smart Computing (BigComp)
- [6] S. Ruder, An overview of gradient descent optimization algorithms*, Insight Centre for Data Analytics, NUI Galway Aylien Ltd., Dublin, 15 Jun 2017
- [7] M. Kaut, S. W. Wallace, Evaluation of scenario-generation methods for stochastic programming, March 2007
- [8] <https://www.wired.com/2016/06/how-google-is-remaking-itself-as-a-machine-learning-first-company/> [22.06.2019]
- [9] P. Lula, Text-mining jaką narzędzie pozyskiwania informacji z dokumentów tekstowych, Akademia Ekonomiczna w Krakowie, Katefra Informatyki, 2005
- [10] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of Tricks for Efficient Text Classification, 6 Jul 2016
- [11] Z. Min-Ling, Z. Zhi-Hua, Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization, 28 August 2006
- [12] S. Fabrizio, Machine learning in automated text categorization, March 2002